Les sites Internet Dynamiques

- Les serveurs et les clients
  - Internet...
  - un jeu de rôle.
  - Mais que fait un client web ?
  - Le serveur web, lui, que fait-il ?

- Statique ou dynamique ?
  - Que fait le serveur d'un site statique ?
  - Le serveur d'un site dynamique, le voltigeur coordinateur.
  -

- Apache, PHP, MySQL …
  - **Le schéma récapitulatif à apprendre par cœur**

- Le travail du webmestre
  - La création du site : le développement et la production (front et back office).
  - Rédiger un cahier des charges de site dynamique.
  -

Les serveurs et leurs clients

Internet …

Internet est un système de mise en relation de différents ordinateurs entre eux. C'est ce que l'on appelle un **réseau** informatique.

Il est constitué de câbles (ou de liaisons électromagnétiques) qui relient les ordinateurs les uns aux autres, et d'un protocole de communication compréhensible par tous ces ordinateurs.

C'est exactement ce qui se passe dans un réseau domestique ou dans le réseau du lycée. Mais à l'échelle mondiale.

Un jeu de rôle …

Mais quand deux ordinateurs communiquent entre eux dans un réseau, en réalité, il y en a un qui joue le rôle de serveur, et l'autre celui du client. C'est même un schéma assez strict dans le « web ».

Le Serveur est un logiciel capable d'écouter ce qui se passe dans le réseau. Il écoute en attendant que quelqu'un s'adresse à lui pour agir. (Le nom « serveur » est aussi donné à l'ordinateur qui exécute un logiciel serveur)

Le client, lui, n'est pas capable d'écouter les conversations qui passent à sa portée. Lui ne sait que demander des choses aux serveurs, et en attendre une réponse appropriée.

C'est exactement ce qui se passe à la terrasse d'un café. Les clients sont à leurs tables, discutent, lisent le journal, boivent leur consommation… l'un d'entre eux veut commander un café. Il lève le doigt et crie à la cantonade « Hep, garçon ! ». Les autres clients ne s'en soucient guerre et continuent leurs occupations sans sourciller. Seul le serveur réagit ! Il s'approche, prend la commande, va demander au bar le café commandé, puis le ramène tout fumant au client qui lui avait demandé. … Le client prend son café, pense parfois à remercier le serveur, et continue ses occupations. Le serveur de retourner à son poste de surveillance ou de s'occuper d'un autre client qui vient de le héler.

Mais que fait un client web ?

Savez-vous que votre **navigateur** Internet Explorer, ou FireFox, ou Safari, est **un client web ?** Maintenant, oui….

Vous savez maintenant que votre navigateur, *parce qu'il est un client*, ne communique pas avec n'importe qui, mais seulement avec des serveurs ! Et il ne fait que lui passer des commandes. On appelle ces commandes des **requêtes**.

Le menu disponible est plutôt sobre. Ce client doit choisir entre :

- des fichiers **HTML**
- des fichiers Javascript
- des fichiers Image
- des fichers Son, Vidéo, et Animations Flash.

Vous savez ce que c'est qu'un fichier HTML. C'est un fichier qui contient du texte, mis en forme grâce à des balises. **Les balises vont être traduites** par le navigateur, et le texte sera affiché à l'écran, mis en forme. Certaines balises permettent d'inclure des images, vidéos et des animations dans ce texte. A chacune de ces balises, le navigateur (le client) fait une requête au serveur pour obtenir le fichier adéquate. A chaque demande, le serveur délivre ses fichiers, les uns après les autres.

Le navigateur et les formulaires. Parfois, la requête est un peu plus compliquée qu'une simple demande de fichier. Le navigateur va demander au serveur un fichier, mais en même temps, il va lui donner des informations complémentaires : celles que vous aurez donné en remplissant un formulaire.

L'ensemble des fichiers HTML, liés entre eux par des liens hypertexte, et disponibles sur le même serveur constituent **un site web.**

Le serveur web, lui, que fait-il ?

Une fois qu'il a reçu la requête d'un client, le but d'un serveur est de lui livrer le fichier attendu.

Dans l'absolu, un serveur web n'est qu'un distributeur de fichiers !

Du moins, c'est ainsi que le client le perçoit. En fait, tout dépend de la nature du **site web sur lequel vous naviguez : statique ou dynamique ?**

Statique ou dynamique ?

La différence entre un site web statique et un site web dynamique se fait un niveau du serveur. Pour le navigateur, le résultat est le même, il reçoit le fichier qu'il a commandé au serveur, un point c'est tout.

Que fait le serveur d'un site statique ?

En réalité, pas grand chose. Ce n'est qu'**une boite à copie** !

Il reçoit en requête le nom d'un fichier. Il va dans **sa réserve de fichiers**, sur son disque dur, trouve celui qui est commandé, en fait une copie, et apporte au client la copie toute fraiche de ce fichier.

A présent, le client peut bien en faire ce qu'il veut de cette copie, le serveur a gardé l'original pour en faire une autre copie, au prochain client qui lui demandera.

Et c'est tout... S'il reçoit dans une requête autre chose qu'un nom de fichier, par exemple des données de formulaire, il va carrément les ignorer. Lui, il fait des copies, et rien d'autre !

Le serveur d'un site dynamique, le voltigeur coordinateur.

Ce serveur là doit distribuer les fichiers d'un site dynamique. C'est une autre histoire qu'une simple copie de fichiers.

En effet, sur son disque dur, **le serveur ne trouve pas les fichiers demandés, mais uniquement leurs plans de construction !**

Hop, ne perdons pas de temps, le serveur se met à la tâche :

1. Il **crée un nouveau fichier**, dans lequel il va construire son œuvre.
2. Sur le plan, on lui dit d'aller chercher du texte qui se trouve d**ans une base de donnée.** Il appelle la base de données, lui demande les infos nécessaires. La base de données lui donne (entre eux, il s'est passé exactement la même conversation qu'entre un client et un serveur). Il met ces données dans son fichier.
3. Puis il est indiqué de **faire un calcul** à partir d'une liste de chiffres qui se trouve **dans la mémoire vive de son ordinateur**, et d'inscrire le résultat à tel endroit. Le serveur va fouiller la mémoire vive, trouve les chiffres, fait son calcul, et l'inscrit dans son fichier au bon endroit.
4. Ensuite, il doit aller chercher dans un fichier texte « cave_a_vin », à la troisième ligne de ce fichier, la phrase qui y est écrite, et l'inscrire à l'identique à la fin de son fichier en chantier.
5. Ah, il y a indiqué que le travail est terminé ! Il « **ferme** » son fichier.

Il ne lui reste plus qu'à livrer son nouveau fichier à son client.

Il ne gardera pas de copie de ce fichier. En fait, ce fichier doit se consommer

très frais. Tous les ingrédients qui le composent son périssables, et il vaut mieux les conserver là ou ils sont stockés. Parfois même, pendant que le serveur s'occupe de ses clients, quelqu'un d'autre vient changer le contenu de la base de données, ou la fameuse troisième ligne du fichier texte. Mais ça, ça ne regarde pas le serveur...

Parfois, dans le plan, il est indiqué d'utiliser **les données fournies dans** la requête du client... ahhhhh, c'est donc à ça que servent **les formulaires !** :-)

Quand un client navigue sur un site dynamique, sans le savoir, **il donne au serveur** non pas le nom d'un fichier à lui livrer, mais **le nom du plan de construction que le serveur doit utiliser.**

Chaque plan de construction est différent. Mais ils sont tous sur le même schéma :

- création d'un nouveau fichier
- liste des tâches à accomplir par le serveur pour construire le contenu.
- instruction de fermeture du fichier, prêt à livrer.

Ce plan de construction s'appelle un script.

Remarquez bien une chose importante : le serveur n'envoie jamais au client web le script, mais le résultat de l'exécution celui-ci. Dans notre exemple de serveur de café, le client ne se retrouve pas avec la recette du café, ou le mode d'emploi de la machine à café, mais avec son café prêt à boire !... avec la sucrette, la touillette et le petit chocolat amer.

Apache, PHP, MySQL …

Les concepts posés, entrons à présent dans le détail des technologies disponibles.

Le client web :

Nous en avons déjà parlé, les clients web les plus couramment utilisés sont les navigateurs. Vous les connaissez, entre autre :
Internet Explorer, Firefox, Safari, Opéra, Netscape...

Le client web se charge :

- d'interpréter le ***HTML***
- d'exécuter de mini-logiciels écrits directement dans les fichiers HTML, ou liés à eux, en ***Javascript.***
- d'assurer un affichage correcte des images
- d'ouvrir le lecteur adéquat pour les sons et les vidéos.

Le serveur web :

Le logiciel de serveur web le plus utilisé dans le monde est ***Apache***. Ce n'est pas le seul, mais c'est le plus connu.

Ce logiciel existe en version Linux et Windows.

C'est un logiciel libre, distribué sous une licence spécifique Apache Licence, les informaticiens peuvent lui ajouter de nombreux modules. Ces modules permettront d'agrandir le nombre d'actions que l'on peut demander au serveur dans nos « plans de construction », dans nos scripts.

L'un de ces modules permet d'écrire nos scripts en « PHP ». Un autre permet de communiquer avec des bases de données, dont « MySQL ».

PHP est un langage de programmation. C'est avec ce « langage » que nous allons traduire nos algorithmes en scripts. Apache lira les commandes PHP exactement comme nous, dans sa même version. Il va les interpréter. ***PHP est donc ce que l'on appelle un langage interprété***.
Par opposition, au langage interprété, on parle de ***langage compilé***. Quand un programmeur crée un script en langage compilé, il doit le confier à un logiciel qui va transformer le script en un code lisible et exploitable par l'ordinateur. Mais cela rend les codes utilisés par la machine incompréhensibles par les programmeurs.
Ce qui fait tout le coté pratique de PHP, c'est qu'il est interprété. N'importe quel programmeur peut lire les scripts et les retoucher. En réalité, le travail d'encodage en langage machine est assuré par Apache.

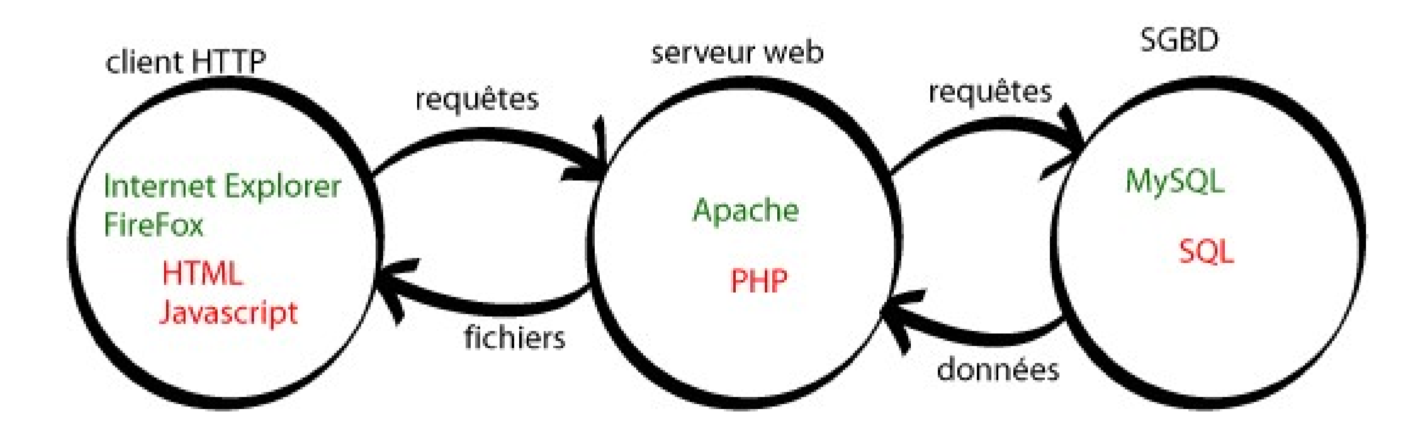
Sur le serveur, on trouve aussi un autre logiciel : le gestionnaire de base de données. MySQL est un logiciel de ce type. C'est aussi un logiciel « libre ». Il

est très connu des webmestres car il est simple à installer sur un serveur, et ses fonctionnalités, réduites mais efficaces dans la majeur partie des cas, consomment peu de ressources.

MySQL est un logiciel distinct de Apache. Quand Apache à besoin de données qui sont gérées par MySQL, ils communiquent entre eux comme le feraient un client et un serveur. Même s'ils se trouvent sur le même ordinateur.

Le schéma récapitulatif à apprendre par cœur :

Notes :
Le **client HTTP** est le nom plus technique de client web. Cela signifie qu'il utilise le protocole de transfert de ficher HTTP (HyperTtext Transfert Protocol)
SGBD signifie Système de Gestion de Base de Données.



Le travail du webmestre

C'est quand les humains s'en mêlent que tout s'embrouille... On va quand même essayer d'y voir clair.

La création du site : le développement et la production.

Nous comprendrons vite que le travail du webmestre vis à vis de son serveur est très différent suivant qu'il crée un site statique ou dynamique.

Dans le cas d'un site statique, le webmestre crée des pages HTML. Il va ensuite les stocker sur le disque dur du serveur.

La phase de développement :

Dans le cas d'un site dynamique, le webmestre va créer les plans de construction, pour les donner au serveur. Seuls certains éléments nécessaires à l'assemblage des fichiers seront donnés par le webmestre :

- les feuilles de style,
- une partie de la structure HTML des documents
- certaines images, ou fichiers annexes.
- Une base de données, vide ou pré-remplie.

Parfois, le webmestre ne crée jamais le contenu des pages. Ce sont d'autre intervenants qui s'en occuperont, grâce à leurs formulaires !

La phase de production :

Il **existe une distinction très nette entre la phase de création du site** (la phase de développement), **et la phase de création des pages web** et du contenu du site : la phase de production.

La production correspond à la période ou le webmester ouvre le site aux différents intervenants qui vont en créer le contenu (nous verrons la notion d'intervenants plus bas).

La plus part du temps, la création du contenu d'un site dynamique va se faire par deux portes d'entrée :

le front office : c'est la partie du site ouverte au publique, constituée de pages d'information ou de formulaires. Les Internautes vont ajouter du contenu au site grâce à des formulaires de contact qui sont accessibles sans mot de passe. Parfois, ils vont remplir le contenu du site sans s'en apercevoir (par l'incrémentation d'un compteur de visite par exemple).

Le back office : c'est la partie privée, **réservée à l'équipe responsable d'édition**. Si un Internaute peut avoir sa propre partie privée (comme dans le cas d'un compte client sur une boutique en ligne), elle ne constitue pas pour autant le « back office ». Ça ressemble beaucoup à un site web, c'est construit pareil, mais ce n'est pas accessible au public. Il faut avoir un identifiant et un mot de passe réservé à l'équipe en charge du site Internet. Par exemple, dans un back

Office, nous allons trouver les page web réservées aux modérateurs d'un forum, celles des outils de publication d'un CMS, etc...

Rédiger un cahier des charges pour un site dynamique

Avec un site dynamique, en plus de **l'arborescence** du site et de son **page à page**, nous allons ajouter des « **fonctionnalités** » à notre site.

Les fonctionnalités, les acteurs, le scénario.

Votre site Internet devient un ensemble de fonctionnalités. Chaque lien dans une page web dynamique devient en réalité une commande de fonctionnalité. Par exemple, celle qui permet d'afficher une page « article », ou celle qui permet d'afficher la page d'accueil .

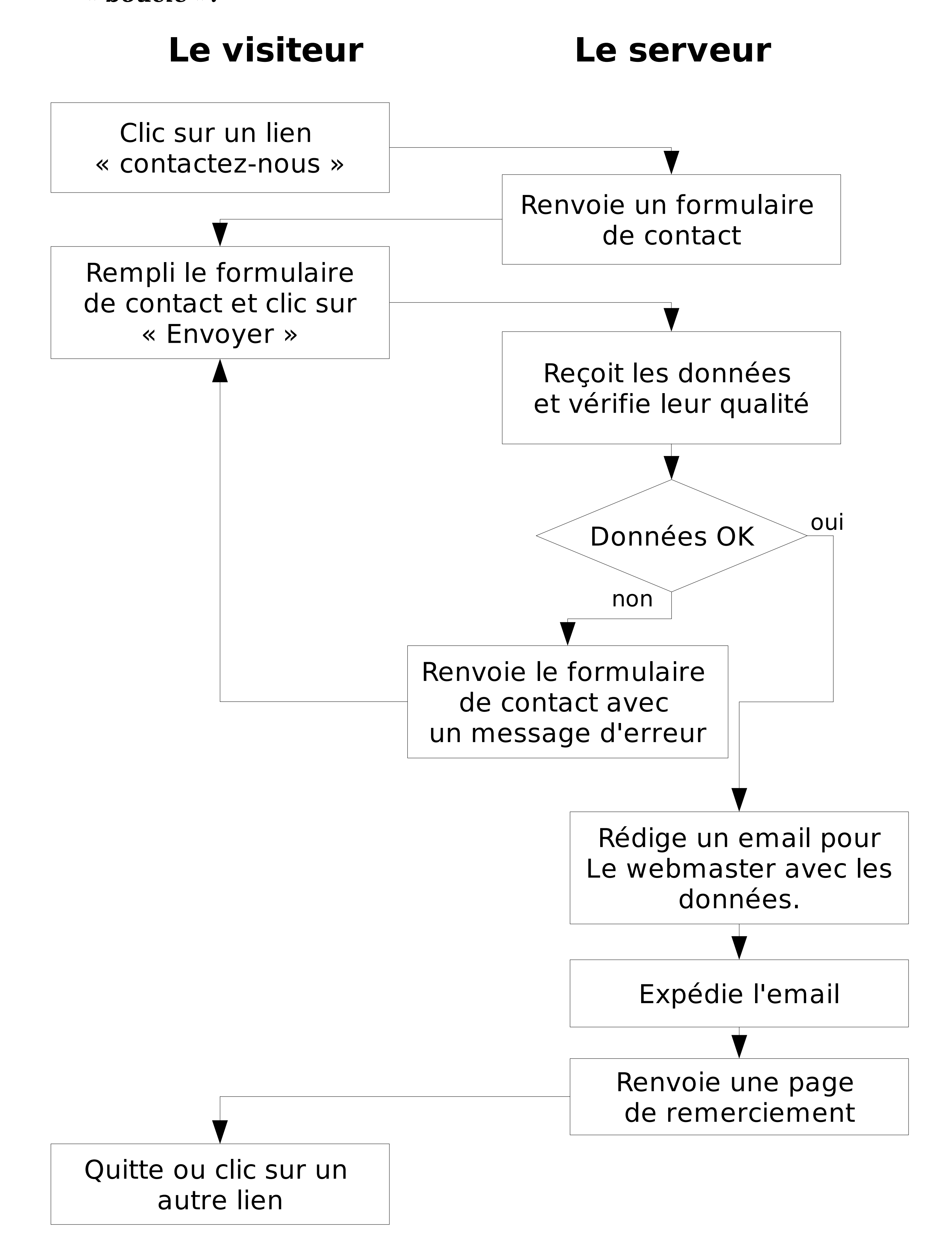
Une site web dynamique va être utile à partir du moment où vous pouvez définir « un acteur » et la tâche qu'il doit accomplir, et la façon de le faire : le scénario.

EXEMPLE :
Le plus simple des **fonctionnalités** courantes sur Internet est le formulaire de contact.
On a 2 acteurs : le visiteur, le serveur.
Le visiteur est un Internaute lambda, sans connaissance particulière en informatique. Il sait lire et écrire. Il souhaite prendre contact avec l'équipe de rédaction de votre site.
Le serveur, c'est notre ordinateur qui héberge notre site web.
Le scénario est le suivant : où qu'il se trouve sur le site, le visiteur peut se rendre par un unique lien hypertexte « contactez-nous » vers une page lui présentant un formulaire. Il devra remplir ce formulaire avec tous les renseignements demandés, puis cliquer sur un bouton « envoyer ». Le serveur recevra ces informations. Il devra d'abord vérifier qu'elle correspondent bien à ce que l'on attend (champs obligatoires par exemple). Si il y a une erreur, le serveur renvoie une page indiquant l'erreur, et le même formulaire de contact à corriger. Si toutes les informations sont correctes, le serveur les inscrit dans un email qu'il envoie à « contact@monsite.fr ». Puis, il envoie au visiteur une page avec inscrit le message suivant « votre message a bien été envoyé, nous vous remercions de votre confiance ». Le visiteur peut continuer sa visite sur le site.

Vous voyez dans cet exemple **les trois phases préparatoires** :
1. définition de la fonctionnalité
2. description des acteurs et caractéristiques
3. résumé du scénario en français courant.

Il est ensuite très simple de réaliser **un schéma de ce scénario**.
- Pour chaque acteur, on crée une colonne d'actions.
- Chaque rectangle correspond à une action.

- Le losange correspond à un choix à faire par l'acteur. De façon très basique, la réponse à ce choix doit être oui/non ou vrai/faux. ***On parle de « condition »***
- Rectangles et losanges sont liés entre eux de façon logique et séquentielle (étape par étape), en suivant le scénario rédigé en français. On peut remonter à un étape antérieure. ***On parle alors de « boucle ».***



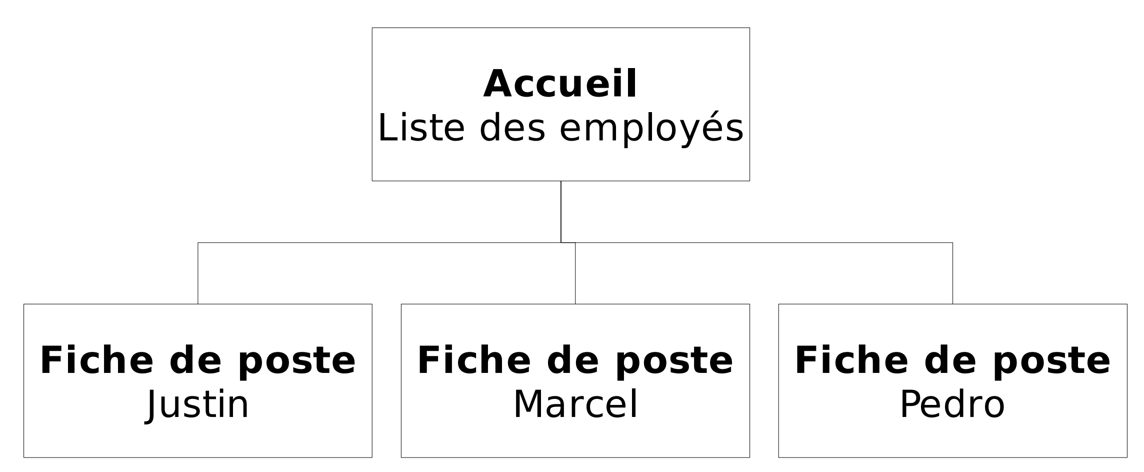
L'arborescence comprimée

Dans un site Internet dynamique, on ne parle plus de pages, mais d'interfaces. Une interface est une page avec un ensemble de commandes de fonctionnalités et des parties, ou des blocs, générés de façon dynamique.

Aussi, notre arborescence de site web ne vas pas schématiser l'ensemble des pages du site, mais l'ensemble des interfaces.

EXEMPLE
Nous devons réaliser un site web avec la fiche de poste de chaque employé. La fiche de poste indique le nom de l'employé, sa photo, ses coordonnées, sa fonction et ses responsabilités au sein de l'entreprise.

Premier cas :
je fais ce site pour **une entreprise de 3 employés** : Justin, Marcel, et Pedro. Voici l'arborescence du site :



On peut même se contenter d'un site statique pour ça, pas besoin de s'embêter avec une base de données !

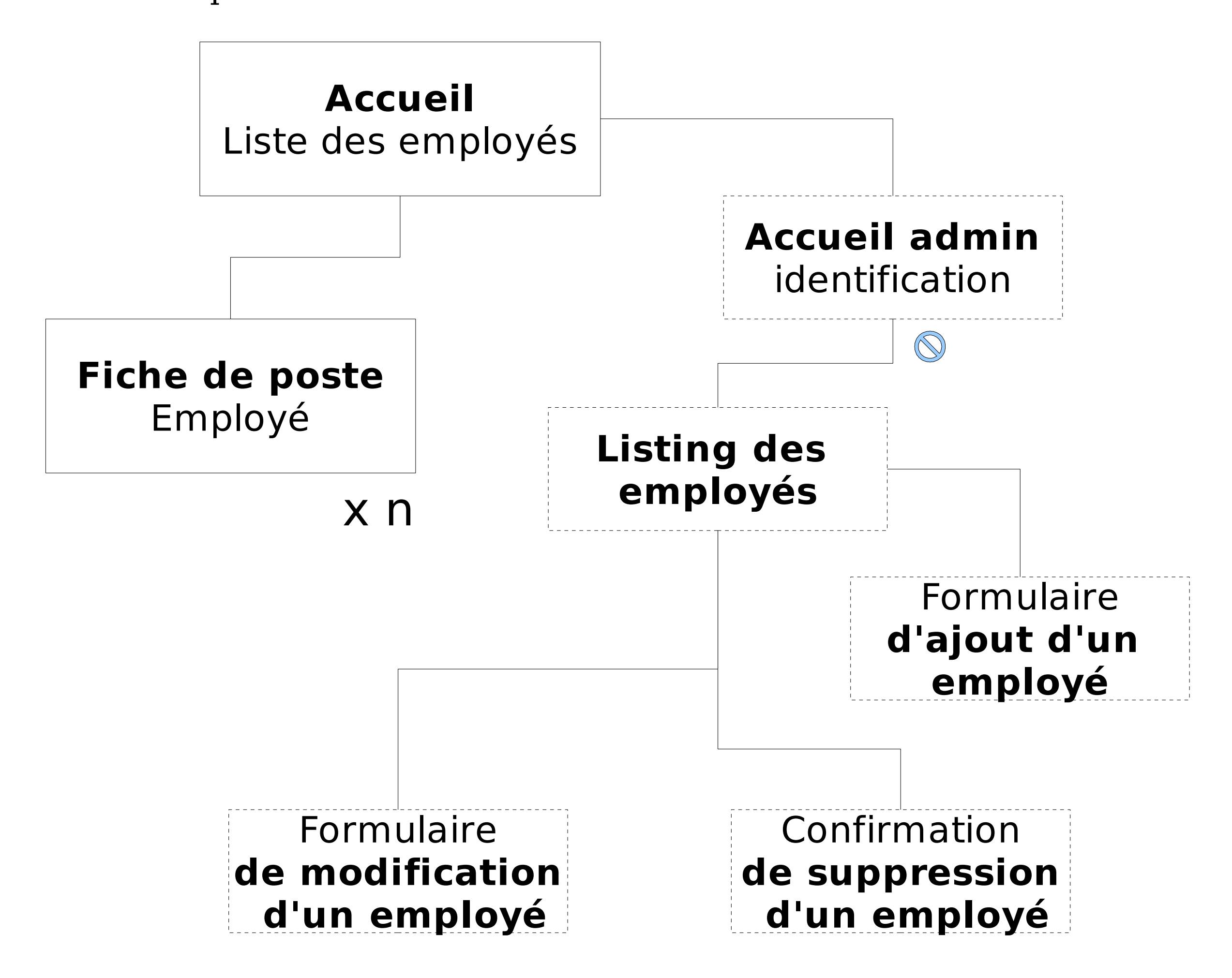
Second cas :
Je fais ce site pour une entreprise de **250 employés** !!!

Nonononon ! Je ne vais pas faire une arborescence de 250 cases avec le nom de chaque employés !
Pire, il est probable qu'avec une telle entreprise, il y ai des employés qui partent et d'autre qui embauchent tous les quatre matin.

Nous allons donc faire **une arborescence d'interfaces,** avec des fonctionnalités propres à un administrateur qui pourra ajouter, modifier et supprimer des fiches de poste, sans l'intervention du Webmaster.

Nous ne créons pas de case pour chaque page du site, mais une pour chaque

« Interface ». La fiche de poste est une interface qui sera générée dynamiquement pour chaque employé. Il n'y a donc plus qu'une seule case « fiche de poste ».



J'ai bien distingué sur mon schéma les interfaces réservées à l'administrateur du site de celles ouvertes au public.
J'y ai mis aussi l'endroit ou va se réaliser le filtre d'accès.

Le page à page ? Non ! l'interface par interface ...

Dans le cas d'un site statique, lorsque vous réalisez un page à page, vous indiquez le texte et l'emplacement des photos que vous avez collecté.

Dans le cadre d'une interface dynamique, nous allons plutôt indiquer le type de données qu'il convient de placer dans la page

EXEMPLE Revenons sur nos fiches de postes.

<u>Cas numéro 1,</u> la petite entreprise avec son site statique. Voici la page de la fiche de poste de Justin :

Fiche de poste de Justin Ploëdec



Fonction : Chef de section

Tél. direct :05 59 140 140

email :justin@laboite.com

Justin est responsable de tout ce que l'on peut lui reprocher. Son objectif principal dans l'entreprise est d'être le bouc émissaire, aussi bien en interne qu'en externe.
Il doit notamment rendre compte des bourdes de ses collègues auprès des clients et du patron.

Cas numéro 2, la très grosse entreprise. Voici l'interface des fiches de poste :

Fiche de poste de <prenom> <nom>

<Photo>

Fonction : <fonction>

Tél. direct : <numero tel>

Email : <email>

<texte multilignes « responsabilité » sur 800 caractères maximum>

C'est exactement la même page, sauf que les éléments dynamiques sont remplacés par « des champs de fusion ». J'ai mis entre < et > les données dynamiques à insérer dans la page.

Pour compléter le dossier, et avoir un bon aperçu de la page, vous pouvez joindre un exemple de la page une fois le contenu dynamique inséré : on retombe alors sur la page avec Justin.

Vous préciserez dans les champs de fusion, le format particulier des données à inserer. Dans notre exemple, le paragraphe « responsabilités » a des spécifications : c'est un paragraphe multi-lignes qui contiendra un maximum de 800 caractères.

Dans les blocs dynamiques, au lieu d'écrire le texte tel qu'il apparaîtra, vous placez des repères explicatifs du genre :

<titre de 200 caractères maximum>
<paragraphe de 3000 caractères maximum>
<date au format JJ/MM/AAAA>

Fin du cours.